REMARKS

Claims 1-28 are pending in the present application. Claims 1, 2, 4, 5, 7-9, 11, 12, 14-16, 18, 19, 20-23, 25, 26 and 28 are amended. Claims 29-31 are added.

Claims 1, 8, 15, and 22 are amended to recite "receiving a request for the original JSP file at a server, wherein the request is sent from the PvC device, and wherein the original JSP file is stored on the server." These features are supported at least on page 9, lines 26-28 and on page 8, lines 5-8 of the current specification.

Claims 2, 9, 16, and 23 are amended to recite "wherein embedding of JSP tags is performed at the server." These features are supported at least on page 10, line 15 to page 11, line 26 of the current specification.

Claims 4, 11, 18, and 25 are amended to recite "storing the transformed JSP file includes storing the transformed JSP file with a filename that is unique to the particular PvC device for which the transformed JSP file is formatted." These features are supported at least on page 10, lines 5-10 of the current specification.

Claims 5, 12, 19, and 26 are amended to recite "the filename has a unique extension for the PvC device for which the transformed JSP file is formatted." These features are supported at least on page 10, lines 5-10 of the current specification.

Claims 7, 14, 21, and 28 are amended to recite "writes the JSP tag to the resultant file embedded in an HTML comment tag, if the tag is a JSP tag." These features are supported at least on page 10, lines 21-22 of the current specification. Claims 20 and 21 are amended to provide proper antecedent basis.

Claims 29, 30, and 31 are added to recite "in response to receiving the request, determining a type of the PvC device based on a header of the request, and locating the original JSP file in the server based on a filename of the original JSP file corresponding to the type of PvC device." These features are supported at least on page 9, line 28 to page 10, line 10 of the current specification.

No new matter is added as a result of the above amendments. Reconsideration of the claims in view of the above amendments and the following Remarks is respectfully requested.

I. 35 U.S.C. § 103(a), Obviousness, Claims 1, 6-8, 13-15, 20-22 and 27-28

The Office Action rejects claims 1, 6-8, 13-15, 20-22, and 27-28 under 35 U.S.C. § 103(a) as being unpatentable over Hawkins et al. (U.S. Patent No. 6,343,318) in view of Ginter et al. (U.S. Patent No. 5,892,900). This rejection is respectfully traversed.

Regarding independent claim 1, the Office Action states:

> Hawkins recites: "The proxy server **180** responds to requests by wireless clients **405** to fetch either web content or messaging information. The proxy server **180** carries most of the burden of bringing the information from the Internet **190**, converting it to wireless client **405** compatible CTP and CML formats, and transferring it to the wireless client **405** over the wireless network" (column 261, lines 17-23) and "The wireless client **405** and the proxy server **180** use a special format for transferring screen **101** contents from the proxy server **180** to the wireless client **405**. This format, named Compact Markup Language (CML), emphasizes compactness over readability and generally uses variable length binary bit fields instead of text to represent options and formatting information" (column 21, lines 33-40). Hawkins further recites: "CGI (Common Gateway Interface) scripts can be supported. CGI scripts are used by the web server **140** to respond to form submissions by browsers and for customizing web content for a particular user. When the browser **104** requests a web document that corresponds to a CGI script, the browser **104** can append text parameters to the end of the base document URL. The proxy server **180** will parse the parameters out" (column 13, lines 44-51) and "Alternatively, the wireless applications can standalone applications access through the browser **104**. The applications can be C programs, JAVA programs, and/or compressed markup language (CML) or HTML pages" (column 9, lines 34-37). Hawkins also recites: "The wireless application **106** represents one of many predefined applications that are stored locally on the wireless communications device **100**" (column 9, lines 15-17).

The Office bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). For an invention to be prima facie obvious, the prior art must teach or suggest all claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974).

Amended independent claim 1, which is representative of claims 8, 15, and 22 with regard to similarly recited subject matter, now recites:

> 1. A method of transforming an original Java Server Page (JSP) file into a pervasive computing (PvC) device specific JSP file, comprising:

> receiving a request for the original JSP file at a server, wherein the request is sent from the PvC device, and wherein the original JSP file is stored on the server;
>
> parsing the original JSP file for JSP tags;
>
> masking the JSP tags;
>
> converting non-masked tags in the original JSP file into PvC device specific format tags;
>
> unmasking the JSP tags; and
>
> storing a transformed JSP file containing the PvC device specific format tags and the JSP tags. (emphasis added)

Neither Hawkins nor Ginter teaches or suggests the feature emphasized above. As discussed in the Abstract, Hawkins teaches a system having a distributed web site. The web site is distributed between a client, a server, and a web server. The client stores a set of predefined applications that correspond to a part of the web site. The applications are formatted according to a first markup language. From the set of predefined applications, the client can generate queries. The server receives the queries and generates new, related queries. The new queries correspond to a second query protocol. The second query protocol is used by the web server. The web server generates responses to the new queries and sends these responses to the server. The responses are formatted according to a second markup language. These responses correspond to the second portion of the web site. The server then converts the responses into new responses that the client can use.

Ginter teaches a system for electronic commerce including secure transaction management and electronic rights protection. As discussed in the Abstract, Ginter teaches secure subsystems used with electronic appliances to provide a distributed virtual distributed environment (VDE) that may enforce a secure chain of handling and control of electronically stored or disseminated information. The virtual distribution environment (VDE) may be used to protect rights of various participants in electronic commerce and other electronic-facilitated transactions.

Neither Hawkins nor Ginter teaches or suggests receiving a request for the original JSP file at a server, wherein the request is sent from the PvC device, and wherein the original JSP file is stored on the server. The Office Action alleges that Hawkins teaches these features at column 261, lines 17-23, and at column 21, lines 33-40, which read, respectively, as follows:

> The proxy server 180 responds to requests by wireless clients 405 to fetch either web content or messaging information. The proxy server 180 carries most of thc burden of bringing the information from thc Internet 190, converting it to wirclcss client 405 compatible CTP and CML formats, and transferring it to the wireless client 405 over the wireless nctwork.
>
> The wireless client 405 and the proxy server 180 use a special format for transferring screen 101 contents from the proxy scrver 180 to the wireless client 405. This format, named Compact Markup Language (CML), emphasizes compactness over readability and generally uses variable length binary bit fields instcad of text to represent options and formatting information.

However, the system of Hawkins docs not receivc a request for the original JSP file at a server, wherein the request is sent from the PvC device and wherein thc original JSP file is stored on the server. **Figure 1** of Hawkins is shown below:

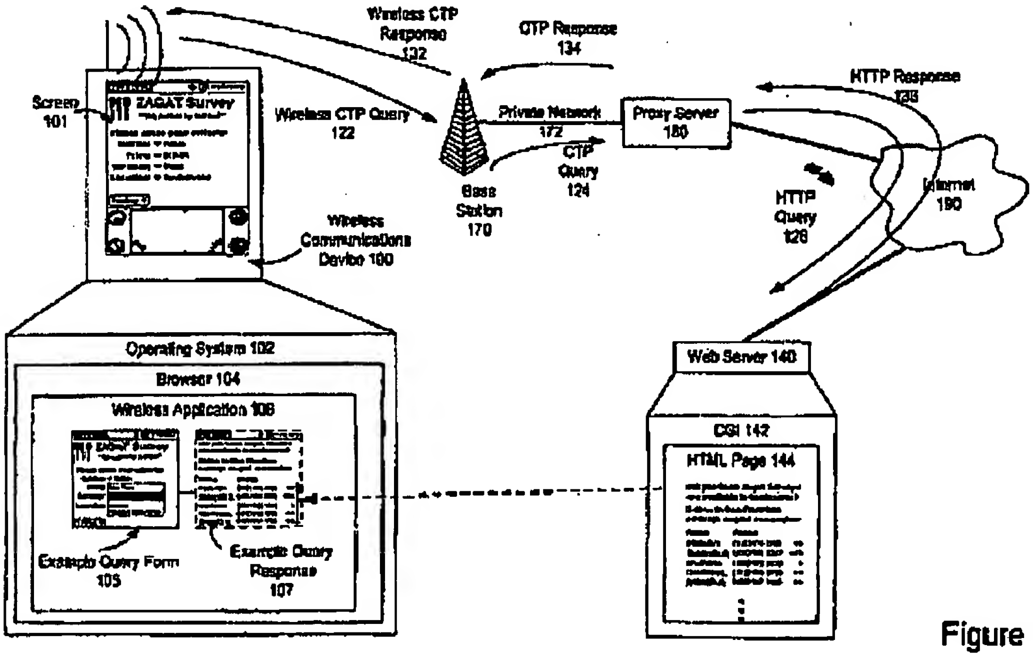


Figure 1

As shown in **Figure 1**, the HTML page is stored on Web server 140. When wireless devicc 100 sends a request for a Web page to proxy server 180, proxy server 180 sends thc query for the Web pagc to Web server 140 via Internet 190. In rcsponse to the request, Wcb server 140 returns the desired Web pagc to proxy server 180, which converts the Web page to a wireless client compatible CTP and CML formats and transfers it back to wireless device **100** (Hawkins, column 261, lines 17-23).

The abovc operation in Hawkins is similar to the prior art mentioned on page 2, lines 17-22 of the current specification, which reads as follows:

> One way to address this problem is to have a transcoding proxy server, or servlet, intercept the JSP and transcode it dynamically, i.e., on the fly, as the JSP is being sent to the PvC device. This option results in extra overhead costs incurred each and every time the JSP is invoked.

Thus, Hawkins merely teaches the prior art method of using a proxy server to transcode the Web page from HTML format to formats that are acceptable by the wireless clients. This is different from the presently claimed invention, which receives the PvC device request at a servcr for an original JSP file, wherein the requested original JSP file is stored on the same server. While Hawkins' proxy server rceeives requests from wireless clients for Web pages, Hawkins' proxy server does not store the Web pages. Rather, Hawkins sends a query to thc Web server, where the Web pages are stored. Therefore, Hawkins does not teach or suggest rcceiving a request for the original JSP file at a server, wherein the request is sent from the PvC device, and wherein the original JSP file is stored on the server, as recited in claims 1, 8, 15, and 22 of the present invention.

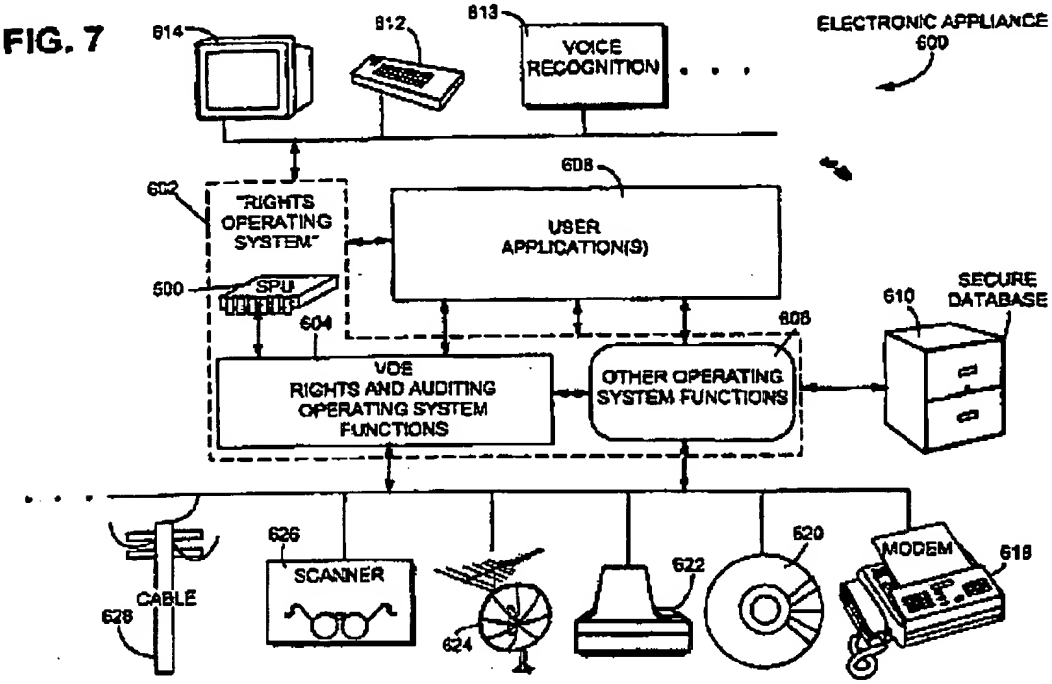
In addition, Hawkins does not teach or suggest parsing the original JSP file for JSP tags. The Office Action alleges that Hawkins teaches these features at column 13, lines 44-51, which reads as follows:

> CGI (Common Gateway Interface) scripts can be supported. CGI scripts are used by the web server **140** to respond to form submission by browsers and for customizing web content for a particular user. When the browser **104** requests a web document that corresponds to a CGI script, the browser **104** can append text parameters to the end of the base document URL. The proxy server **180** will parse the parameters out of the URL.

However, in the above section, Hawkins merely teaches parsing the text parameters that are appended to the end of the base document URL. The text parameters are not the same as original JSP files, in that the text parameters are part of a request scnt by thc wireless client for retrieval of a Web page. The text parameters are not original JSP files that are stored on the server. Therefore, Hawkins fails to teach or suggest parsing the original JSP file for JSP tags, as recited in claims 1, 8, 15, and 22 of the present invention.

Ginter also does not teach or suggest these features. **Figure 7** of Ginter, which gives an example of an electronic appliance, is shown below:

FIG. 7



As shown in **Figure 7**, Ginter teaches an electronic appliance that may be any kind of electronic device, such as a computer, a T.V. set top control box, a pager, a telephone, a sound system, or a video reproduction system. Electronic appliance **600** may communicate with the outside world through cable **628**. Electronic appliance **600** includes a rights operating system **602**, which manages appliance **600** and SPU **500** by controlling their hardware resources. The rights operating system supports application **608**, which may be a program loaded by a user (Ginter, column 60, line 58 to column 61, line 34).

Thus, Ginter's VDE focuses on a rights operating system that runs on an electronic appliance. The rights operating system does not receive a request from a PvC device, because it is already running on the PvC device. In addition, the electronic appliance does not store an original JSP file. Rather, the electronic appliance stores

VDE objects or secured files for secure processing. Therefore, Ginter does not teach receiving a request for the original JSP file at a server, wherein the request is sent from a PvC device, and wherein the original JSP file is stored on the server.

Neither Hawkins nor Ginter teaches or suggests masking and unmasking JSP tags. The Office Action alleges that while Hawkins fails to teach masking and unmasking of tags, Ginter teaches masking of tags at column 150, lines 35-37, which reads as follows:

> This site-specific key masks a validation tag that may be derived from a cryptographically strong pseudo-random sequence by the SPE 503.

In the above section, Ginter teaches masking a validation tag. According to column 149, lines 10-12 of Ginter, a validation tag is a reference tag in a Document Type Definition (DTD) that describes a correct DTD requested by the secure processing element. The validation tag in Ginter is different from a JSP tag, in that a validation tag is a tag that is included in a <u>DTD document</u> for describing a DTD file to be requested by a SPE. This is contrary to a JSP tag, which is a tag parsed from an <u>original JSP file</u> that designates JSP components, such as a Java bean or applets. Therefore, while Ginter teaches masking a tag, Ginter does not teach or suggest masking and unmasking a JSP tag, as recited in claims 1, 8, 15, and 22 of the present invention.

Furthermore, the Office Action alleges that it would have been obvious to a person of ordinary skill in the art to combine the masking of tags as taught by Ginter with the transformation of files for pervasive computing devices as taught by Hawkins in order to "maintain the integrity, availability, and/or confidentiality of such information and processes related to such use", as suggested by Ginter at column 1, lines 13-15. Applicants respectfully disagree.

Hawkins is not concerned with receiving a request for an original JSP file at a server from a PvC device, wherein the original JSP file is stored on the server, parsing the original JSP file for JSP tags, and masking those tags. Rather, Hawkins forwards the request from a wireless device from a proxy server to a Web server, where the requested Web page is stored. Hawkins then converts the content from HTML to CML at the proxy server before transferring it to the wireless device. Thus, Hawkins merely teaches a prior art method that the present invention seeks to avoid.

While Ginter teaches masking a validation tag, Ginter is not concerned with masking a JSP tag that is parsed from an original JSP file stored on a server. A person of ordinary skill in the art would not have been led to combine the masking of tags in Ginter with the transformation of content in Hawkins to mask JSP tags that are parsed from an original JSP file stored on the server, since neither Hawkins nor Ginter even teaches or suggests JSP tags. In addition, even if a person of ordinary skill in the art were to combine the teachings of Hawkins and Ginter, the resulting combination would still not be receiving a request at a server for an original JSP file, wherein the request is sent from the PvC device, and wherein the original JSP file is stored on the server, parsing the original JSP file for JSP tags, and masking the JSP tags.

To the contrary, the resulting combination would be receiving a request for a Web page from a wireless device at a proxy server, forwarding the request from the proxy server to a Web server, where the Web page is stored, masking a validation tag in a DTD, and converting the content of the Web page from a HTML format to a CML format at the proxy server before returning the content to the wireless device.

In view of the above, Applicants respectfully submit that neither Hawkins nor Ginter, either alone or in combination, teaches or suggests the features of claims 1, 8, 15, and 22. At least by virtue of their dependency on claims 1, 8, 15, and 22 respectively, neither Hawkins nor Ginter teaches or suggests the features of dependent claims 6, 7, 13, 14, 20, 21, 27, and 28. Accordingly, Applicant respectfully requests the withdrawal of the rejection of claims 1, 6-8, 13-15, 20-22, 27, and 28 under 35 U.S.C. § 103(a).

In addition, neither Hawkins nor Ginter, either alone or in combination, teaches or suggests the specific features as recited in dependent claims 6, 7, 13, 14, 20, 21, 27 and 28. The Office Action alleges that these claims are rejected for fully incorporating the deficiencies of their base claims. However, neither Hawkins nor Ginter teaches or suggests such features. For example, with regard to amended claim 6, which is representative of claims 13, 20, and 27 with regard to similarly recited subject matter, neither Hawkins nor Ginter teaches or suggests determining if a tag is encountered, if the tag is encountered, determining if the tag is an HTML tag, and if the tag is not an HTML tag, identifying the tag as a JSP tag.

While Hawkins teaches identifying HTML tags, Hawkins does not teach or suggest identifying the tag as a JSP tag if the tag is not an HTML tag. Rather, Hawkins teaches using a special format, Compact Markup Language (CML), for transferring content. Therefore, not only does Hawkins fail to teach identifying a tag as JSP tag, Hawkins teaches away from identifying a tag as a JSP tag by teaching the use of CML tags for transferring content.

Ginter also does not teach identifying a tag as JSP tag. There is no teaching or suggestion of JSP tags in Ginter. Ginter merely teaches VDE objects that are used to protect rights of those who creates electronic content, provides electronic products, provide electronic credit and currency, uses electronic content, and described by electronic information (column 4, lines 14-28). Ginter is not concerned with identifying any tag as a JSP tag. Therefore, Ginter also does not teach or suggest the features of claims 6, 13, 20, and 27 of the present invention.

As to dependent claim 7, which is representative of claims 14, 21, and 28 with regard to similarly recited subject matter, neither Hawkins nor Ginter teaches if a tag is not encountered, writing the JSP file content to a resultant file, if a tag is a HTML tag, writing the HTML tag to the resultant file, and if the tag is a JSP tag, writing the JSP tag to the resultant file embedded in an HTML comment tag.

As discussed above in arguments presented for claim 6, Hawkins does not teach identifying a tag as a JSP tag if the tag is not an HTML tag. As Hawkins does not teach a tag or a JSP tag, it follows that Hawkins would not teach that if a tag is a JSP tag, the tag is written to the resultant file embedded in an HTML comment tag. To the contrary, at column 22, lines 20-30, Hawkins merely contrasts the use of CML tags with the use of HTML tags. Thus, as Hawkins only teaches writing CML tags, as opposed to HTML or JSP tags, to the resultant file, Hawkins fails to teach or suggest writing JSP tag embedded in an HTML comment tag to a resultant file. Therefore, Hawkins does not teach or suggest the features of claims 7, 14, 21, and 28 of the present invention.

Ginter also does not teach if a tag is a JSP tag, writing the tag to the resultant file embedded in an HTML comment tag. As discussed above in arguments presented in claim 6, Ginter merely teaches VDE objects for protecting rights of electronic content. Thus, Ginter does not teach or suggest anything regarding identifying a JSP tag or writing the tag to a resultant file. Therefore, Ginter also does not teach or suggest the features of claims 7, 14, 21, and 28 of the present invention.

Thus, in addition to their dependency on claims 1, 8, 15, and 22, Applicants respectfully submit that neither Hawkins nor Ginter, either alone or in combination, teaches or suggests the specific features of claims 6, 7, 13, 14, 20, 21, 27, and 28. Accordingly, Applicants respectfully request the withdrawal of rejections to claims 6, 7, 13, 14, 20, 21, 27, and 28 under 35 U.S.C. § 103(a).

II. 35 U.S.C. § 103(a), Alleged Obviousness, Claims 2-3, 9-10, 16-17 and 23-24

The Office Action rejects claims 2-3, 9-10, 16-17, and 23-24 under 35 U.S.C. § 103(a) as being unpatentable over Hawkins et al. (U.S. Patent No. 6,343,318) in view of Ginter (U.S. Patent No. 5,892,900) and further in view of Judson (U.S. Patent No. 6,185,586 B1). This rejection is respectfully traversed.

Regarding claims 2 and 3, the Office Action states:

> Hawkins discloses transforming a Java proxy server file application into a pervasive computing device compatible file, where the server will parse specific elements out during the conversion process. Hawkins also discloses storing the transformed file. Hawkins and Ginter disclose masking as described above. Hawkins and Ginter fail to disclose masking by use of comment tags. Judson teaches the use of comments tags to mask. Judson recites: "Preferably, the information object is masked by an HTML comment tag, which may include other HTML tags nested therein to format the information in the object" (column 3, lines 2-3).

As discussed above in arguments presented for claims 1, 8, 15, and 22, Hawkins and Ginter fail to teach or suggest receiving a request for an original JSP file at a server, wherein the request is sent from a PvC device and wherein the original file is stored on the server. Judson also fails to teach or suggest these features.

Judson teaches a method of display as a user of the Internet using a client machine during an Internet transaction. As discussed in the Abstract, the method locally stores information content served during idle periods when the user's connection to the network is live. During a given Internet transaction, the information content is retrieved and displayed to provide entertainment or information as the user waits for the Internet transaction to be completed.

However, Judson does not teach or suggest receiving a request for an original JSP file at a server, wherein the request is sent from a PvC device and wherein the original file is stored on the server. Judson teaches a system that is implemented at the client browser at column 2, lines 60 to column 3, line 8, which reads as follows:

> The invention is preferably implemented in a computer having a processor, an operating system, a graphical user interface and a HTTP-compliant browser. In such case, the novel and advantageous features of the invention are achieved using a first means, responsive to activation of a link from a web page, for retrieving an information object masked within the web page, and a second means for displaying information from the information object on the graphical user interface as the browser establishes the link. Preferably, the information object is masked by an HTML comment tag, which may include other HTML tags nested therein to format the information in the object. This enables the support of complex "mini" web pages that are displayed and accessible to the viewer during otherwise non-productive periods when the browser is busy processing links to other documents or web sites.

In the above section, Judson teaches embedding information objects within HTML comment tags at the client browser in order to display mini web pages when the browser is busy. Thus, Judson only teaches a mechanism that is implemented at the client browser. Consequently, Judson fails to teach or suggest receiving a request for the original JSP at the server, wherein the request is sent from a PvC device and the original JSP file is stored on the server, as recited in claims 1, 8, 15, and 22 of the present invention.

In addition, Hawkins, Ginter, and Judson fail to teach embedding the JSP tags into HyperText Mark-up Language (HTML) comment tags, wherein the embedding step is performed at the server. The Office Action admits that neither Hawkins nor Ginter teaches or suggests these features. However, the Office Action alleges that Judson teaches these features at column 3, lines 2-3, which reads as follows:

> Preferably, the information object is masked by an HTML comment tag, which may include other HTML tags.

While Judson teaches embedding information objects, including other HTML tags, within a Web page, the embedding step is performed at the client browser when a link is activated from the Web page. Judson does not teach or suggest embedding JSP tags into HTML comment tags at the server.

Judson is only concerned with masking information objects embedded within the web page, such that the information objects may be retrieved and displayed while the browser is busy processing links to other documents and websites. Judson is not concerned with embedding JSP tags at the server. Therefore, Judson does not, and would not, teach or suggest embedding JSP tags into HTML comment tags at the server, as recited in claims 2, 9, 16, and 23 of the present invention.

In addition, the Office Action alleges that it would have been obvious to one of ordinary skill in the art to combine the transformation of files for PvC devices by masking tag as taught by Hawkins and Ginter with the comment masking of Judson, so that "the information is preferably "hidden" within the web page using a HTML comment tag (Judson, column 2, lines 58-59).

However, there is no teaching or suggestion in Hawkins or Ginter to mask JSP tags by embedding JSP tags into HTML comment tags at the server. Hawkins is only interested in converting HTML tags to CML tags and Ginter is only interested in masking a validation tag in a DTD document. Neither reference makes any suggestion to mask JSP tags. In addition, there is no teaching or suggestion in Judson to embed JSP tags into HTML comment tags at the server. Judson is only interested in masking information objects at the client browser for display in a graphical user interface of the web browser. Therefore, a person of ordinary skill in the art would not be led to modify or combine the teachings of Hawkins, Ginter, and Judson to embed JSP tags into HTML comment tags at the server.

Furthermore, even assuming, arguendo, that Hawkins, Ginter, and Judson were properly combinable, the resulting combination would not be embedding JSP tags into HTML comment tags at the server. Rather, the resulting combination would be embedding information objects into HTML comment tags at the client browser, and converting the HTML tags into CML tags at the proxy server. Therefore, even if a person of ordinary skill in the art were to combine Hawkins, Ginter, and Judson, the resulting combination would still not be the same as the presently claimed invention.

As to dependent claims 3, 10, 17, and 24, Judson does not teach or suggest removing HTML comment tag identifiers from the HTML comment tags in which the JSP tags are embedded. Judson does not mention anything about unmasking or removing

HTML comment tag identifiers, as Judson is only interested in hiding information within a Web page in a HTML comment tag, such that when a link is activated, the information may be formatted as a mini web page to be displayed to the user. Judson is not interested in unmasking the information object. Judson simply retrieves the object identified in the comment tag and displays it in the Web page.

In view of the above, Applicants respectfully submit that Hawkins, Ginter, and Judson do not teach or suggest the features of claims 2-3, 9-10, 16-17, and 20-21 of the present invention. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 2-3, 9-10, 16-17, and 20-21 under 35 U.S.C. § 103(a).

III. 35 U.S.C. § 103(a), Alleged Obviousness, Claims 4-5, 11-12, 18-19 and 25-26

The Office Action rejects claims 4-5, 11-12, 18-19 and 25-26 under 35 U.S.C. § 103(a) as being unpatentable over Hawkins et al. (U.S. Patent No. 6,343,318) in view of Ginter (U.S. Patent No. 5,892,900) and further in view of Ramaley et al. (U.S. Patent No. 6,585,177 B1). This rejection is respectfully traversed.

Regarding claims 4 and 5, the Office Action states:

> Hawkins discloses transforming a Java proxy server file application into a pervasive computing device compatible file, where the server will parse specific elements out during the conversion process. Hawkins also discloses storing the transformed file. Hawkins and Ginter disclose masking as described above. Hawkins and Ginter fail to disclose storing with a unique file name or file extension. Ramaley discloses "Assign Unique Identifier Comprising Fixed String and Unique Instance Number" at reference **620** in Fig. 6. Ramaley teaches the use of unique file naming.

As described above in arguments presented for claims 1, 8, 15, and 22, Hawkins and Ginter fail to teach or suggest receiving a request for an original JSP file at a server, wherein the request is sent from a PvC device and wherein the original file is stored on the server. Ramaley also fails to teach or suggest these features.

As discussed in the Abstract, Ramaley teaches:

> A computer-implemented process for simulating, in an HTML-formatted primary file, certain user-expected characteristics of an electronic document containing an embedded file. In response to initiating a save operation for a primary file to be saved as an HTML document, a file embedded within the primary file is detected. In turn, a support file is created for storage, in a separate

> storage location and physically removed from the primary file, to logically represent the embedded file. In addition, a current file list, which is associated with the primary file, is created and saved to the storage mechanism, typically the hard disk drive for the computer. When the HTML-formatted primary file is opened in a work session, the editor can open a link to each support file identified by the file list. In response to conducting a save operation for a primary file, an inquiry can be conducted to determine if a prior file list is associated with the primary document. If a prior file list is located, then the current file list is compared to the prior file list. This comparison operation is completed to determine whether support files identified by the prior file list are no longer identified by the current file list. The support files not identified by the current file list, but identified in the prior file list, are deleted from the hard disk of the computer because the corresponding embedded files have been deleted by edit operations for the primary file.

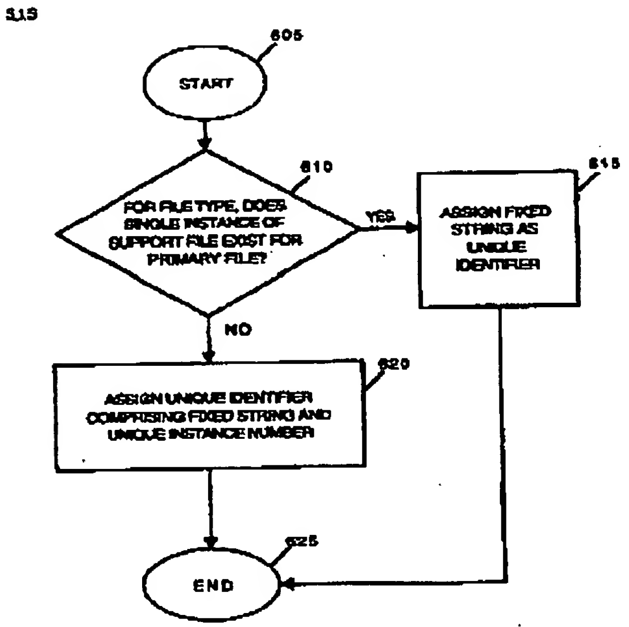
Ramaley does not teach receiving a request for an original JSP file at a server, wherein the request is sent from a PvC device and the original JSP file is stored on the server. At column 3, lines 12-28, Ramaley teaches the following:

> Although HyperText Markup Language ("HTML") files contain links to electronic files, rather than embedded files, the present invention can simulate the characteristics exhibited by an electronic document having one or more embedded files. For example, users of typical desktop productivity tools, such as word processing or spreadsheet programs, have the expectation that opening an electronic file containing an embedded file will result in the opening of that embedded file. For a corresponding HTML-formatted file, which can not contain an embedded file because of the inherent limitations of the HTML file format, the present invention achieves this desirable characteristic by saving a primary file having an embedded file to a storage mechanism, such as a hard disk drive, and saving each embedded file as a linked support file in a known location on the storage mechanism.

Thus, Ramaley teaches saving a primary file and embedded file to a hard disk drive of a user desktop, as opposed to a server. Therefore, Ramaley fails to teach the features of claims 1, 8, 15, and 22 of the present invention.

In addition, Ramaley does not teach storing the transformed JSP file with a filename that is unique to the particular PvC device for which the JSP file is formatted (claims 4, 11, 18, and 25) or that the filename has a unique extension for the PvC device for which the JSP file is formatted (claim 5, 12, 19, and 26). The Office Action alleges that Ramaley teaches these features in **Figure 6**, **element 620**, which is shown below:

Fig. 6


515
605
START
610
FOR FILE TYPE, DOES SINGLE INSTANCE OF SUPPORT FILE EXIST FOR PRIMARY FILE?
YES
615
ASSIGN FIXED STRING AS UNIQUE IDENTIFIER
NO
620
ASSIGN UNIQUE IDENTIFIER COMPRISING FIXED STRING AND UNIQUE INSTANCE NUMBER
625
END

As shown in **Figure** 6 and at column 6, lines 35-55, Ramaley teaches assigning a unique identifier to a support file based on a combination of a fixed string and an unique instance number. For example, the first instance of the support file type can be identified by a fixed string and the instance number 001 (the number 1 preceded by a padding of one or more zeroes). The second instance of this support file type includes the identical fixed string and a different instance number, typically in numerical order, such as 002, and so forth for other instances of this support file type.

Instead of a filename that is unique to a particular PvC device for which the JSP file is formatted, the filename in Ramaley is composed of an identical fixed string and a different instance number for each instance. Thus, the filename in Ramaley is not unique to a particular PvC device. Rather, Ramaley's filename is only unique to the instance of support file type. Therefore, Ramaley does not teach the features of claims 4, 11, 18, and 25 of the present invention.

For similar reason set forth above, Ramaley also does not teach a filename that has a unique extension for the PvC device for which the JSP file is formatted. Ramaley merely teaches a filename that has unique extension for the instance of the support file type. Ramaley does not teach a filename that has unique extension for the PvC device, because Ramaley is only storing support files in a single user's workstation for each particular instance. Ramaley is not storing files for each PvC device. Therefore, Ramaley also does not teach a filename that has a unique extension for the PvC device for which the JSP file is formatted, as recited in claims 5, 12, 19, and 26 of the present invention.

The Office Action alleges that it would have been obvious to one of ordinary skill in the art at the time the invention was made to add the file naming of Ramaley to the transformation of files for pervasive computing devices by masking tag as taught by Hawkins and Ginter to provide the benefit of "placing information in a primary file that provides a cue" (Ramaley, column 3, lines 7-8). Applicants respectfully disagree.

At column 3, lines 5-10, Ramaley teaches placing information in a primary file that provides a cue to an editing program, such as a web page editor, that a particular file associated with that primary file should be treated as either embedded or linked content. Thus, Ramaley is concerned with placing information in the primary file to identify a unique associated file for a program, rather than with naming a file in such a way to identify a unique PvC device. Therefore, a person of ordinary skill in the art would not have been led to combine the teachings of Hawkins and Ginter with Ramaley to identify a unique PvC device.

Even assuming, arguendo, that Hawkins, Ginter, and Ramaley were properly combinable, the resulting combination would still not include the features of storing the transformed JSP file with a filename that is unique to the particular PvC device for which the JSP file is formatted, wherein the filename has a unique extension for the PvC device for which the JSP file is formatted. Rather, the resulting combination would be storing a support file with an embedded file with a file name that is unique to the support file type instance, wherein the filename has a unique extension for the particular instance of the associated file.

In view of the above, Applicants respectfully submit that Hawkins, Ginter and Ramaley do not teach or suggest the features of claims 4-5, 11-12, 18-19, and 25-26. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 4-5, 11-12, 18-19, and 25-26 under 35 U.S.C. § 103(a).

IV. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: 10/7/04

Cathrine Kinslow
Reg. No. 51,888
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 367-2001
Attorney for Applicants

CK/im